part of paper #12

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, with sufficient postage, in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on

December 20, 2002
Date of Deposit

Debra M. Haag
Name

Debra M. Haag
Signature

Case No. 10432/28

Commissioner for Patents
Washington, D.C. 20231

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Housel et al.

Serial No: 09/694,734
Filed: 10/23/2000
For: Directing pages to a selected output destination of a printing system

Examiner: Nolan Jr., Charles H.
Group Art Unit: 2854

DECLARATION OF PRIOR INVENTION UNDER 37 CFR § 1.131

Sir:

I, Edward M. Housel being of sound mind hereby state:

1. That I am a co-inventor of the above identified patent application.

2. Prior to July 28, 2000, I and my co-inventor, Robert Brummond, had completed our invention as described and claimed in the subject application in this country, as evidenced by the following:

a. Prior to July 28, 2000, I and my co-inventor, Robert Brummond, as evidenced by Exhibit A, conceived a method and apparatus of printing, comprising storing a medium in at least one input source prior to printing; supporting a user's selection of an output destination, among two or more output destinations, for any sheet of a print job in the at least one input source prior to the printing; and determining a pattern of media feeds for each output set of the print job to achieve a desired appearance characteristic for the output set or a desired assembly of the sheets of the output set.

b. Prior to July 28, 2000, as evidenced by Exhibits B and C, two KDK commands were written into a Postscript Document Structuring Conventions parser at my request that support a user's selection of an output destination, among two or more output destinations, for any sheet of a print job in at least one input source prior to printing, the KDK commands being Document Structuring Convention commands executable by a printing system wherein a medium is stored in the at least one input source prior to printing, the printing system determining a pattern of media feeds for each output set of the print job to achieve a desired appearance characteristic for the output set or a desired assembly of the sheets of the output set.

3. The dates deleted from Exhibits A, B, and C are all prior to July 28, 2000.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Edward M Housel

Edward M. Housel

12/20/2002

Date

Exhibit A

DOCKET # 79879

ASSIGNED DATE:

Title:

Method of Selectively Directing Pages To a Plurality of Output Trays

Inventor:
Ed Housel , revised

Field Of Invention:

This invention relates generally to digital printing systems anu duplicating systems having at least two paper trays, and at least two output destinations. Further, this system may be loaded with different media types in each paper tray. This system is also capable of producing collated sets of sheets which are deposited in one of these output destinations.

Background:

Digital printing systems which have two paper supplies, and two output destinations are known (i.e. Kodak IS92sp). Currently, these products allow a customer to select only one output destination for the output sets for a customer job. However, these products allow for all of their input paper trays to be used for any sheet of the output set. Currently, printing systems and copiers which produce sets of collated output do not allow the operator to choose an output destination for individual pages of the output set.

Some products support special features which allow two output destinations to be used during a customer job. These products typically use the top exit of their system as a "purge" tray. For these jobs, the system determines that some of the media loaded in one of the paper trays is not needed by the job, but must be fed through the system so that the next output set does not use it. These systems calculate how many sheets must be "purged" using information supplied to it. These systems do not provide the flexibility to let a requestor choose when to route a page to this "purge" tray. Finally, these systems do not allow printing on the pages which are sent to the "purge" tray.

Summary of Invention:

The present invention is intended to address one or more of the above mentioned problems. Briefly summarized according to one aspect of the present invention a printing system provides a means for a user to instruct the system to choose an unique output destination for any sheet of the output set. The user can choose to have the media printed or unprinted. The user can enable the system to provide jam recovery

emh

instructions to the operator. Finally, the user can request that the system determine a the pattern of media feeds for each set of output in order to achieve desired aesthetic characteristics for the output sets. Using all this information, the system creates a pattern of media feed instructions which are repeated for each output set.

Brief Description of Drawings:

Detailed Description:

Fig 1 shows a block diagram of a printing system. The CPU (17) in this diagram is the computer which accepts customer jobs from the network (10) and controls all aspects of printing them on the printer (18). The jobs may come from another computer (11) which creates electronic descriptions of pages for the printer. The user of computer (11) can request the features from their computer screen (13) which might display a user interface screen as shown in Fig 4.

Referring to Fig 1, another source of jobs comes from the computer (16) which is connected directly to the CPU (17). This computer (17) scans existing pages and produces electronic versions of the pages for printing. The CPU (17) handles all aspects of the printing process for the printer (18). The CPU (17) may be physically implemented using one or more computer systems, but they all work collectively to control the printing process.

Referring to Fig 1, the CPU (17) receives jobs. An operator of the printer (18) uses the computer screen (14) to check view job status, and check how the customer jobs were set up. Among the many user interface screens displayed for the operator is a screen as shown in Fig 4. Additional screens allow the operator to view the attributes of the media loaded in the paper trays (32), (33), and (34) which are shown in Fig 2.

Fig 2 shows a printing system which has three paper trays (32), (33), and (34). The computer screen (31) is the same as the computer screen (14) shown in Fig 1. This computer screen includes a keyboard and mouse for user input (not shown). The printer (38) is the same printer (18) shown in Fig 1. The output destinations (35), (36), and (37) might be trays, or they could represent finishing devices such as staplers, stacker, folders, binders etc. For the purposes of this invention, the nature of them is not important.

Fig 3 shows several examples of media having a repeating pattern of collated pages. The printer operator might load one of these medias in a paper tray (32), (33), and (34). These would be loaded as repeating groups of either (51), (61) or (71). For instance, the operator might load (71) which is two part tabs. This media has a repeating sequence containing 2 unique sheets. As loaded, there might be one hundred groups of two part tabs (71) loaded in paper tray1 (32). Paper tray2 (33) and paper tray3 (34) might contain letter media.

Fig 4A shows a user interface screen (101) which might be display on computer screen (13). From this screen, the user can enable all aspects of the present invention. The user can select a media using pull down menu (102). For the media selected in (102), the attributes are shown in text box (104). The user can select a duplex or simplex using the pull down menu (105). The user can select a job exit using pull down menu (106). The options for pull down menu (106) are: <JobExit>, out1, out2, out3, out1+, out2+, out3+. "out1" is the abbreviation for output destination1 (35). "out1" is the abbreviation for output destination1 (36). "out3" is the abbreviation for output destination3 (37). The "+" sign denotes that the user wants jam recovery assistance for these pages and media identified. The user types a list of page numbers and the keyword "last" on the entry line (103). This list also accepts additional key words such as "B" as shown in column (117) of text box (115) shown in Fig 4. After typing a line on (103), the user selects an option from pull down menu (107). This puts the typed information into textbox (115), and then erases all text from the line (103).

Fig 4B shows the user interface screen (101) as it would look after a user typed a list of page numbers on entry line (103). This user also selected a new media called "TabB". This media is described in the text box (104). Finally, the user selects the option "Insert" from the pull down menu (107). Once this is done, the user interface screen (101) would be as shown in Fig 4C. By repeating these steps, the user would eventually populate the user interface screen (101) as shown in Fig 4D

An example of an output set which could be generated by the system is shown in Fig 6. This output would be generated from twelve input pages, in a job which has already requested the <JobExit> to be the output destination2 (36) in another user interface screen which is not shown. The operator loads two part tab media (71)into paper tray1 (32). Paper tray2 already contains "letter" media. The user types lines (108) and (109) on the user interface shown in Fig 4. The media name is "TabB" as shown in column (116) of the text box in Fig 4. The user wants printing all pages. The user also enables jam recovery messaging using the "+" appended to the desired output destination as shown in column (119). Fig 6 shows one output set. The pages in group (91) are sent to the <JobExit> which is output destination2 (36). Pages in group (91) are actually stacked on top of each other in the output destination. Fig 6 shows them next to each other for purposes of clearly identifying the "TabB" media sheets. The first pages delivered are pages one, two and three (92). These are printed on "letter" media. Input page 4 is printed on the first "TabB" media (93). Pages five, six and seven are printed on "letter" media (94). Input page 8 is printed on the second "TabB" media (95). Pages nine, ten and eleven are printed on "letter" media (96). Input page 12 is printed on the second "TabB" media (97). The media (98) is sent to output destination1 (35) with page 13 printed on it. For this example, the media repeats every two sheets, but the job request repeats every four sheets. The MediaExit pattern for this job is given below in table 1.

Table 1: Example of a MediaExit Pattern

Page	Printed On?	Output Destination
4	True	<JobExit>
8	True	<JobExit>
12	True	<JobExit>
13	True	Output Destination1

A second example of an output set which could be generated by the system is shown in Fig 5. This output would be generated from twelve input pages, in a job which has already requested the <JobExit> to be the output destination2 (36). The operator loads five part tab media (51)into paper tray1 (32). Paper tray2 already contains "letter" media. The user types lines (113) and (114) on the user interface shown in Fig 4. The media name is "TabA" as shown in column (116) of the text box in Fig 4. The user wants printing all pages. The user also enables jam recovery messaging using the "+" appended to the desired output destination as shown in column (119). Fig 5 shows one output set. The pages in group (81) are sent to the <JobExit> which is output destination2 (36). Pages in group (81) are actually stacked on top of each other in the output destination. Fig 5 shows them next to each other for purposes of clearly identifying the "TabA" media sheets. The first pages delivered are pages one, two and three (82). These are printed on "letter" media. Input page 4 is printed on the first "TabA" media (83). Pages five, six and seven are printed on "letter" media (84). Before printing page 8, the system sends an unprinted "TabA" media (88) to output destination1 (35). The system built this page feed command using a special balanced algorithm as shown in Fig 8. This algorithm was enabled by the user with the "B" in the page list from line (114) of the text box (115) in column (117) shown in Fig 4. Input page 8 is printed on the third "TabA" media (85). Pages nine, ten and eleven are printed on "letter" media (86). Before printing page 12, the system sends an unprinted "TabA" media (88) to output destination1 (35). The system built this page feed command using a special balanced algorithm as shown in Fig 8. Input page 12 is printed on the fifth "TabA" media (87). For this example, the media repeats every two sheets, but the job request repeats every four sheets. The MediaExit pattern for this job is given below in table 2.

Table 2: Example of a Balanced MediaExit Pattern

Page	Printed On?	Output Destination
4	True	<JobExit>
8	False	Output Destination1
8	True	<JobExit>
12	False	Output Destination1
12	True	<JobExit>

Referring to Fig 4, the user could type line (110). Unlike the previous examples, the user does not enable jam recovery messages. This allows a page from the job to be sent to the output destination1 (35). Since this page is printed, the may have wanted part of their job handled differently by the printer operator. For example, this might be an instruction sheet for the operator describing what to do with the rest of the printed output. It could also be a status page.

Referring to Fig 4, the user could type lines (111), (112) and (120). These would print on "color" media on pages seven and fifteen. At the end of each set, two unprinted sheets of "color" media would be sent to the output destination1 (35). Then one unprinted sheet of "color" media would be sent to output destination2 (36).

Fig 7 shows a flow chart for the method of the jam recovery assistance. In the event of a paper jam in the printer (38), the system displays messages instructing the operator to remove all paper in the system paper path (201). Once this is complete as tested in decision box (202), the system checks if the user wants jam recovery assistance for any media used by the job (203). If assistance was enabled, then the system displays a message showing where the printing system is in its MediaExit Pattern (204). The operator would have to open the paper supply to verify the top sheet of media was correct for the current set. Then the machine would continue to print the output set.

Fig 7B shows an alternative method for jam recovery. In the event of a jam in the printer (38), the system displays messages instructing the user to clear the jam (205). Once the jam is cleared, as tested in the decision box (206), the system checks if the user wants jam recovery assistance for any media in the job (207). If assistance was enabled, then the system displays a message telling the user to discard the partially printed set, and to verify that the paper supply is correct to start printing the next set (208).

Fig 8 shows a flow chart for the method of Balanced MediaExit Pattern Determination. This method would be enabled by a user by typing the "B" in column (117) of the text box (115) in figure 4. Before balancing, the MediaExit pattern would be as shown in table 3 below.

Table 3: Example of an Unbalanced MediaExit Pattern

Page	Printed On?	Output Destination
4	True	<JobExit>
8	True	<JobExit>
12	True	<JobExit>
last	False	Output Destination1
last	False	Output Destination1

Referring to Fig 8, for each sheet feed done by the printer (38), this method would be invoked. If the media being fed is not the MediaExit media (221), then "exit". The word "exit" means the method is complete and no further action is needed for this sheet feed. Process step (222) examines the MediaExit pattern, keeping a count of the number of times the "<JobExit>" is found. The MediaExit pattern of table 3 has 3 occurrences of "<JobExit>". Step (223) checks for an invalid MediaExit Pattern. The test (223) checks if the printer is at the start of a set of output prints. At the start of a set, step (225) initializes a number of items needed later. The LowAdjust value may be set from 0 to 1. Experiments have shown good results with 0.9. "Route to Job Exit" means that the engine sends the first MediaExit media from to the "<JobExit>". On a subsequent sheet feed, test (226) exits if there is only 1 "<JobExit>" request in the MediaExit pattern. Process step (227) examines the MediaExit pattern, keeping a count of the number of times any exit is used. The MediaExit pattern of table 2 has 5 occurrences. The test (228) exits if all the required media feeds for this set have been done. The test (229) executes the process step (230) causing the last request for the MediaExit media to be routed to the "<JobExit>". So far, this method has attempted to allocate the first and last entries in the MediaExit pattern for "<JobExit>" pages. Process step (231) calculates "Target1" and "Target2" to be used in tests (232) and (233). These determine if one of the remaining "<JobExit>" entries should be used for the current media feed as shown in process step (234). If not the sheet is routed to the alternate exit specified in the MediaExit Pattern. Finally, process step (235) increments the count of MediaExit media feeds. The system effectively generates table 2 from table 3 using this method.

Referring to Fig 9, an alternate user interface (301) is shown. This user interface is almost identical to Fig 4 (101). The differences are the addition of the "Define Pattern" pull down menu (304) and the MediaExit Pattern column (303), and the removal of the "Exit" pull down menu (106) and the "exit" column (119) in Fig 4. All other elements function the same as in Fig 4. The "Define Pattern" pull down menu

(304) has three options: "Define Pattern", "Edit Pattern" and "Remove Pattern". "Define Pattern" brings up the user interface shown in Fig 10 (351). In Fig 9, the user has entered a page list on text line (103). They have selected media called "TabA" in pull down menu (102). The attributes of the media are shown in text box (104). Finally, they have selected "Duplex" in the pull down menu (105).

Referring to Fig 10, the system knows that the user has asked for "TabA" on three page numbers. The user interface screen (351) is populated as shown. The text box (352) describes the media being defined. The "exit" pull down menus (353) and (354) allow the user to select up to two exits in addition to the current "<JobExit>". The exit names which populate these pull down menus are limited to those available on the printer (38). The user changes "Exit2" pull down menu (353) to "stacker". The columns (359), (360), (361), (362), (363), and (367) contain 4 rows (355), (356), (357) and (358). The first row (355) "Create New Insert" starts out with no check boxes checked. The user checks the second and fourth check box. This results in the row (355) shown in Fig 11. At print time, this results in 2 extra sheets of unprinted "TabA" being used by each set of printed output. The user clicks the radio button in row (357) column (360). This automatically turns off the radio button in row (356) column (360). The system also causes the radio button in row (356) and column (362) to become selected. In this way, the system enforces the rule that the row (356) must always have the same number of selections as the number of times "TabA" has been requested for the job. The user clicks the radio button in row (357) column (362). This automatically turns off the radio button in row (356) column (362). The system also causes the radio button in row (356) and column (363) to become selected. When the user selects the "OK" button (364) on Fig 11, the user interface is shown in Fig 12. The "Cancel" button (366) in Fig 10 returns to the screen shown in Fig 9 which means no pattern was set for the "TabA" media. The "Reset" button (365) in Fig 10 restores the screen to the original state shown in Fig 10. This example was shown with twelve columns ending with column (367) in Fig 10. There could be as few as two columns, and there is no upper limit on the number of columns. The upper limit on the number of "Exit" rows (357) and (358) to display is dependent on the number of exits available on the print engine (38).

Referring to Fig 12, the text line (302) appears in the text box (305). Column (303) shows the "+P" which identifies the request as having a MediaExit pattern associated with it. To remove this pattern, the user can use the "Remove Pattern" from the pull down menu (304). "Edit Pattern" allows changes to a previously set up pattern. The pattern resulting from this example is similar to that shown in Table 2.

Claims:

(TBD)

Exhibit B

Table 2.4-2: 9110 Specific Document Structuring Conventions

%%Key	Description / Allowed Values	Supported By
	Redacted	
%KDKInserts: 3.0 pageNum of 0 no longer would cancel a possible front cover. lastpage no longer cancels a possible back cover. 4.0 allows 0.N (was 1.N) 4.0 allows new finisher destination 4.0 allows "jobexit" as well as syntax which allows multiple output exits per 1 media.	{ mediaName + { {"output(" + "top" | "stacker" | "stapler" | "bookmaker" | "postprocessor" | "top1" | "stapler1" + "~~jobexit~~ + ")"}$_{0.1}$ + {{pageNum | PageRange | "lastpage" }$_{1.M}$ }$_{1.P}$ }$_{0.N}$ → use job setting *The insert is placed after the page indicated.* *Note: Using KDKInsert without KDKBody should be considered an invalid use of KDKInsert. Results may differ from expected.* *Note: Multiple inserts are allowed for a page. They are honored in the order in which they appear.* *Note: pageNum may be 0. This results in an insert before page 1 of the job.* *Note: P is an integer multiple of the "order" of a pre-collated media.* *Note: Since these are page level exceptions, PDL requests are honored unless there is a conflict on a certain page.*	1580, IS70cpII, DHV1
	Redacted	

Exhibit C

Table 2.4-2: 9110 Specific Document Structuring Conventions

%%Key	Description / Allowed Values	Supported By

Redacted

VH

KDKPeOutput ~~Media~~ 4.0 for RIP to Store support and Ordered Media support	{"usepdl"}$_{0.1}$ | { {"top" | "stacker" | "stapler" | "bookmaker" | "postprocessor"}$_{1.1}$ + {pageNumber} $_{1.1}$ }$_{0.N}$ * *Sets the finishing options starting with the pageNumber specified. The options are in effect until the next options/pageNumber specified or PDL changes it..* * * *This interacts with KDKChaptersAreSets.* * ~~staple 1, top 1~~ OK	DHV1

Redacted